AN AUGMENTED ITERATIVE METHOD FOR LARGE LINEAR TOEPLITZ SYSTEMS

Jacob Benesty, M. Mohan Sondhi, and Tomas Gaensler
Bell Labs, Lucent Technologies
700 Mountain Avenue
Murray Hill, NJ 07974, USA
e-mail: {jbenesty,mms,gaensler}@bell-labs.com

ABSTRACT

Efficiently solving a large linear system of equations, $\mathbf{Ax} = \mathbf{b}$, is still a challenging problem. Such a system appears in many applications in signal processing, especially in some problems in acoustics where we deal with very long impulse responses, i.e. $\mathbf{x}$ is long. In this paper, we show how to efficiently use the so-called basic iterative algorithms when the matrix $\mathbf{A}$ is Toeplitz, symmetric, and positive definite. We also propose an improved version that converges much faster than some other iterative methods. We present some simulations and compare the new method to the conjugate gradient algorithm.

1 Introduction

Efficiently solving a large linear system of equations, $\mathbf{Ax} = \mathbf{b}$, is still a challenging problem. Such equations appear in many applications in signal processing, especially in some problems in acoustics where we deal with very long impulse responses, i.e. $\mathbf{x}$ is long. In the rest of this paper, we assume that $\mathbf{A}$ is a square matrix of size $n \times n$, which is symmetric and positive definite (s.p.d.).

There are basically two different approaches to solve such a linear system. The first is to use a direct method such as Gaussian elimination. It requires $O(n^3)$ arithmetic operations and the obtained solution $\mathbf{x}$ is exact [1]. Unfortunately, this technique becomes impractical when n is large. In contrast to the direct methods are the iterative methods [2]. These methods generate a sequence of approximate solutions $\{\mathbf{x}_k\}$ with an arithmetic complexity per iteration much smaller than that of a direct method, since only matrix-vector multiplications and vector additions are needed. Further, if the matrix $\mathbf{A}$ is sparse, its sparsity can be fully exploited. Of course, the value of an iterative method depends on how quickly the iterates $\mathbf{x}_k$ converge. Ideally, an iterative method should have a much lower complexity than that of a direct method, should converge fast, and its convergence rate should be independent of the size n.

Roughly speaking, there are two classes of iterative methods: the basic iterative methods (which we discuss briefly in Section 2) and the subspace iteration methods of which the conjugate gradient algorithm (CGA) is a particular case [3]. The methods in the first category are, in general, less complex than those in the second, but converge much slower.

The CGA is one of the best known iterative techniques for solving sparse symmetric positive definite linear systems. Briefly, the method utilizes orthogonal projection onto a space called the Krylov subspace [4], [5], defined by the vectors $\mathbf{r}_0$, $\mathbf{Ar}_0$, ..., $\mathbf{A}^k\mathbf{r}_0$, where $\mathbf{r}_0 = \mathbf{Ax}_0 - \mathbf{b}$ is the residual for the initial vector $\mathbf{x}_0$. Eventually, for $k = n$ the Krylov vectors span the whole space, so the CGA gives the exact solution after at most n iterations. Therefore, the method can also be seen as a direct solution method. A nice feature of the CGA is that it is parameter-free.

In this paper, we show how to apply the so-called basic iterative algorithms to a Toeplitz system using $O(n \log n)$ operations per iteration. Then, we propose an improved version. We also compare this method, by simulations, to the CGA and show that it is less complex and converges faster in general.

2 Basic Iterative Algorithms

In this section we give a brief review of basic iterative methods for solving the linear system

$$\mathbf{Ax} = \mathbf{b}. \tag{1}$$

The linear system (1) may be rewritten as follows

$$(\mathbf{A} + \mathbf{T})\mathbf{x} = \mathbf{b} + \mathbf{Tx}, \tag{2}$$

where $\mathbf{T}$ is an appropriately chosen preconditioning matrix. We assume that $(\mathbf{A} + \mathbf{T})$ is nonsingular. From (2) we deduce the straightforward iterative algorithm [1]:

$$\mathbf{x}_{k+1} = \mathbf{Hx}_k + \mathbf{v} \tag{3}$$

where

$$\mathbf{H} = (\mathbf{A} + \mathbf{T})^{-1}\mathbf{T} \tag{4}$$

is the iteration matrix and

$$\mathbf{v} = (\mathbf{A} + \mathbf{T})^{-1}\mathbf{b}. \tag{5}$$

For this algorithm, $\mathbf{T}$ can in general be singular. The matrix $\mathbf{T}$ must be such that $\mathbf{y}_k = \mathbf{Tx}_k$ is simple to compute, $(\mathbf{A} + \mathbf{T})$

is easy to invert, and, of course, $\mathbf{x}_k$ converges to $\mathbf{A}^{-1}\mathbf{b}$. The Jacobi and Gauss-Seidel procedures are typical members of this family.

It can be shown that a necessary and sufficient condition for convergence of the basic iterative methods is that

$$\rho(\mathbf{H}) < 1, \tag{6}$$

where $\rho(\mathbf{H})$ is the spectral radius of the matrix $\mathbf{H}$. The spectral radius is defined as

$$\rho(\mathbf{H}) = \max_{\lambda \in S(\mathbf{H})} |\lambda| \tag{7}$$

where $S(\mathbf{H})$ is the set of all eigenvalues of $\mathbf{H}$ (i.e., the spectrum of $\mathbf{H}$). If the matrix $\mathbf{T}$ is symmetric, it follows from (6) that the condition for convergence is

$$\lambda_{\min}(\mathbf{A}^{-1}\mathbf{T}) > -1/2. \tag{8}$$

Theorem 1: If $\mathbf{A}$ is s.p.d., then a sufficient (but not necessary) condition for convergence of the iterative method is that $\mathbf{T}$ is also s.p.d.

Proof: Suppose that $\mathbf{T}$ is s.p.d., then:

$$\mathbf{H} = (\mathbf{I} + \mathbf{T}^{-1}\mathbf{A})^{-1}. \tag{9}$$

Since $\mathbf{T}$ and $\mathbf{A}$ are both s.p.d., it follows that $\lambda_i(\mathbf{T}^{-1}\mathbf{A}) > 0,\ i = 1, 2, ..., n$. Hence $\max|\lambda_i(\mathbf{H})| < 1$, which is the condition for convergence. $\diamond$

We summarize the algorithm:

• Initialization:

$\mathbf{x}_0$, initial guess

$\mathbf{v} = (\mathbf{A} + \mathbf{T})^{-1}\mathbf{b}$

• Iterations $k = 0, 1, 2, ...$:

$\mathbf{y}_k = \mathbf{T}\mathbf{x}_k$

$\mathbf{x}_{k+1} = (\mathbf{A} + \mathbf{T})^{-1}\mathbf{y}_k + \mathbf{v}$

The convergence of this method can be greatly improved in many ways. In the following, we show how that can be done in the particular case when $\mathbf{A}$ is a Toeplitz matrix. That will be our main contribution in this paper.

3 Application to a Toeplitz System

The linear system of (1) where the matrix $\mathbf{A}$ is symmetric and Toeplitz, arises in many problems in signal processing. [A matrix is Toeplitz if the entries on each diagonal are identical.] Such a matrix $\mathbf{A}$ is completely specified by the n elements of its first row

$$\mathbf{A}(1,:) = \begin{bmatrix} a_0 & a_1 & \cdots & a_{n-1} \end{bmatrix}.$$

To exploit this property, we will make use of circulant matrices.

A circulant matrix $\mathbf{C}$, is a special case of a Toeplitz matrix, in which each successive row contains the elements of the row above shifted one step to the right, with the last element wrapped around to become the first. A circulant matrix is symmetric if its first row has the following symmetry:

$$\mathbf{C}(1,:) = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_2 & c_1 \end{bmatrix}.$$

It is well known that a circulant matrix is easily decomposed as follows: $\mathbf{C} = \mathbf{F}^{-1}\Lambda\mathbf{F}$, where $\mathbf{F}$ is the Fourier matrix and Λ is a diagonal matrix whose elements are the Fourier transform of the first column of $\mathbf{C}$. Hence a circulant matrix may be inverted using FFT (Fast Fourier Transform) in $O(n \log n)$ operations. A matrix-vector product of the form $\mathbf{C}\mathbf{x}$ can be computed efficiently as well.

Next we show how to apply these properties to the basic iterative method.

3.1 Simple Version

Suppose we choose $\mathbf{T}$ to be a Toeplitz matrix derived from the given Toeplitz matrix $\mathbf{A}$, such that its first row is given by

$$\mathbf{T}(1,:) = \begin{bmatrix} \alpha & a_{n-1} & a_{n-2} & \cdots & a_2 & a_1 \end{bmatrix},$$

where α is a parameter to be specified. Note that the last $(n-1)$ elements of $\mathbf{T}(1,:)$ are just those of $\mathbf{A}$ in reverse order. Hence, it is clear that with this choice of $\mathbf{T}$, the matrix $\mathbf{C}_1 = \mathbf{A} + \mathbf{T}$ is circulant for any α. The choice of α will control the convergence of the algorithm. A sufficient (but not necessary) condition for convergence is to choose this parameter in such a way that $\mathbf{T}$ has strong diagonal dominance (i.e., $\alpha > 2\sum_{i=1}^{n-1}|a_i|$), since in this case, $\mathbf{T}$ will be positive definite. However, in practice this choice for α is very conservative, and a much smaller value may be used. We do not yet know the optimal value for this parameter.

Since $\mathbf{C}_1 = \mathbf{A} + \mathbf{T}$ is circulant, the computation of $\mathbf{v}$ in (5) can be performed very efficiently. If we could also compute $\mathbf{T}\mathbf{x}_k$ efficiently, then we would have an efficient way of computing the iteration of (3). What makes this possible is the observation that the matrix $\mathbf{C}_2 = \begin{bmatrix} \mathbf{A} & \mathbf{T} \\ \mathbf{T} & \mathbf{A} \end{bmatrix}$ is also circulant. Therefore, $\mathbf{T}\mathbf{x}_k$ can be computed efficiently by noting that $\mathbf{T}\mathbf{x}_k = \begin{bmatrix} \mathbf{I}_{n\times n} & \mathbf{0}_{n\times n} \end{bmatrix}\mathbf{C}_2\begin{bmatrix} \mathbf{0}_{n\times 1} \\ \mathbf{x}_k \end{bmatrix}$.

The algorithm may now be summarized as follows:

• Initialization:

$\mathbf{x}_0$, initial guess

$\mathbf{C}_1 = \mathbf{A} + \mathbf{T} = \mathbf{F}_n^{-1}\Lambda_1\mathbf{F}_n$

$\mathbf{C}_2 = \begin{bmatrix} \mathbf{A} & \mathbf{T} \\ \mathbf{T} & \mathbf{A} \end{bmatrix} = \mathbf{F}_{2n}^{-1}\Lambda_2\mathbf{F}_{2n}$

$\mathbf{v} = \mathbf{C}_1^{-1}\mathbf{b}$

• Iterations $k = 0, 1, 2, ...$:

$\mathbf{y}_k = \begin{bmatrix} \mathbf{I}_{n\times n} & \mathbf{0}_{n\times n} \end{bmatrix}\mathbf{C}_2\begin{bmatrix} \mathbf{0}_{n\times 1} \\ \mathbf{x}_k \end{bmatrix}$

$\mathbf{x}_{k+1} = \mathbf{C}_1^{-1}\mathbf{y}_k + \mathbf{v}$

Per iteration, this algorithm requires 2 FFTs of size n and 2 FFTs of size $2n$, while the CGA requires 4 FFTs of size $2n$. That means that we save more than 25% of operations per iteration. Unfortunately, this algorithm converges in general slower than the CGA. We show in the next section how the convergence rate can be greatly improved.

3.2 Improved Version

The convergence rate, r, of an iterative algorithm is defined as $r = -\log_{10}\rho(\mathbf{H})$, so that 10^{-ri} bounds the decay of the

approximation error after i iterations. If we take $\mathbf{T} = \mathbf{A}$, then $\mathbf{H} = \frac{1}{2}\mathbf{I}$ and $\rho(\mathbf{H}) = 1/2$. That would give a convergence rate of $r \approx 0.3$ which is very good, and more importantly, independent of the size n. Obviously we cannot take $\mathbf{T} = \mathbf{A}$, for that would require computation of $\mathbf{A}^{-1}$ to compute $\mathbf{v}$ in (5). However, we can approach this convergence rate and yet have an efficient algorithm, as shown below.

Note that in most applications the main diagonal and the few adjacent diagonals of $\mathbf{A}$ are strongly dominant. That suggests that we choose the preconditioning matrix such that a few diagonals on either side of the main diagonal are identical to those of $\mathbf{A}$. In this way it would become close enough to $\mathbf{A}$ to improve the rate of convergence. We can use this idea, and yet utilize the properties of circulant matrices, by augmenting the sizes of the matrices. Thus choose $p \ll n$, and define a symmetric $(n+p) \times (n+p)$ Toeplitz matrix $\mathbf{A}'$ in which the first row is obtained by extending the first row of $\mathbf{A}$ as follows:

$$\mathbf{A}'(1,:) = \left[\begin{array}{ccccccc} \mathbf{A}(1,:) & a_p & a_{p-1} & \cdots & a_1 \end{array} \right].$$

Analogously to the case discussed in the previous section, define a matrix $\mathbf{T}'$ such that $\mathbf{A}' + \mathbf{T}'$ is circulant. That is, define the first row of $\mathbf{T}'$ to be

$$\mathbf{T}'(1,:) = \left[\begin{array}{ccccccccc} \beta & a_1 & a_2 & \cdots & a_p & a_{n-1} & \cdots & a_1 \end{array} \right].$$

Note that the first p diagonals on either side of the main diagonal are identical in $\mathbf{A}'$ and $\mathbf{T}'$. The symmetric Toeplitz matrix $\mathbf{A}'$ may be decomposed as follows:

$$\mathbf{A}' = \left[\begin{array}{cc} \mathbf{A} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^T & \hat{\mathbf{A}}_p \end{array} \right],$$

The sub-matrix $\hat{\mathbf{A}}_p$ is a principal submatrix of $\mathbf{A}$ of order p. The matrix $\mathbf{T}'$, of size $(n+p) \times (n+p)$, is also symmetric and Toeplitz and is such that $\mathbf{A}' + \mathbf{T}' = \mathbf{C}_1'$ is circulant. We can also decompose

$$\mathbf{T}' = \left[\begin{array}{cc} \mathbf{T} & \mathbf{T}_{12} \\ \mathbf{T}_{12}^T & \hat{\mathbf{T}}_p \end{array} \right],$$

where $\hat{\mathbf{T}}_p$ is a principal submatrix of $\mathbf{T}$ of order p, and $\mathbf{T}$ is the $n \times n$ symmetric Toeplitz matrix whose first row consists of the first n terms of $\mathbf{T}'(1,:)$. Thus

$$\mathbf{T}(1,:) = \left[\begin{array}{ccccccccc} \beta & a_1 & a_2 & \cdots & a_p & a_{n-1} & \cdots & a_{p+1} \end{array} \right].$$

Now, let us consider the following augmented linear system:

$$\begin{aligned} (\mathbf{A}' + \mathbf{T}') \left[\begin{array}{c} \mathbf{z} \\ \mathbf{z}' \end{array} \right] &= \left[\begin{array}{cc} \mathbf{T} & \mathbf{A}_{12} + \mathbf{T}_{12} \\ \mathbf{0}_{p\times n} & \mathbf{0}_{p\times p} \end{array} \right] \left[\begin{array}{c} \mathbf{z} \\ \mathbf{z}' \end{array} \right] \\ &+ \left[\begin{array}{c} \mathbf{b} \\ \mathbf{0}_{p\times 1} \end{array} \right]. \end{aligned} \tag{10}$$

Theorem 2: Suppose that $(\mathbf{A}' + \mathbf{T}')$ is nonsingular then the solution of the linear system (10) is: $\mathbf{z} = \mathbf{x}$ and $\mathbf{z}' = \mathbf{x}' = -(\hat{\mathbf{A}}_p + \hat{\mathbf{T}}_p)^{-1}(\mathbf{A}_{12}^T + \mathbf{T}_{12}^T)\mathbf{x}$, where $\mathbf{x}$ is the solution of the linear system (1).

Proof: Obvious by developing (10). $\diamond$

Then, we deduce from (10) the iterative algorithm

$$\left[\begin{array}{c} \mathbf{x}_{k+1} \\ \mathbf{x}'_{k+1} \end{array} \right] = \mathbf{H}' \left[\begin{array}{c} \mathbf{x}_k \\ \mathbf{x}'_k \end{array} \right] + \mathbf{v}' \tag{11}$$

where

$$\mathbf{H}' = (\mathbf{A}' + \mathbf{T}')^{-1} \left[\begin{array}{cc} \mathbf{T} & \mathbf{A}_{12} + \mathbf{T}_{12} \\ \mathbf{0}_{p\times n} & \mathbf{0}_{p\times p} \end{array} \right] \tag{12}$$

and

$$\mathbf{v}' = (\mathbf{A}' + \mathbf{T}')^{-1} \left[\begin{array}{c} \mathbf{b} \\ \mathbf{0}_{p\times 1} \end{array} \right]. \tag{13}$$

This iterative algorithm converges if and only if $\rho(\mathbf{H}') < 1$. This algorithm may look somewhat complicated, but it requires the same complexity as the simple version given in Section 3.1, except that n has been increased to $n+p$.

We summarize the algorithm:

• Initialization:

$\mathbf{x}_0$, $\mathbf{x}'_0$, initial guesses

$\mathbf{C}_1' = \mathbf{A}' + \mathbf{T}' = \mathbf{F}_{n+p}^{-1} \Lambda_1' \mathbf{F}_{n+p}$

$\mathbf{C}_2' = \left[\begin{array}{cc} \mathbf{A}' & \mathbf{T}' \\ \mathbf{T}' & \mathbf{A}' \end{array} \right] = \mathbf{F}_{2(n+p)}^{-1} \Lambda_2' \mathbf{F}_{2(n+p)}$

$\mathbf{v}' = \mathbf{C}_1^{'-1} \left[\begin{array}{c} \mathbf{b} \\ \mathbf{0}_{p\times 1} \end{array} \right]$

• Iterations $k = 0, 1, 2, ...$:

$$\mathbf{y}_k = \left[\begin{array}{cc} \mathbf{I}_{n\times n} & \mathbf{0}_{n\times n} \end{array} \right] \mathbf{C}_2' \left[\begin{array}{c} \mathbf{0}_{n\times 1} \\ \mathbf{x}'_k \\ \mathbf{x}_k \\ \mathbf{x}'_k \end{array} \right]$$

$$\left[\begin{array}{c} \mathbf{x}_{k+1} \\ \mathbf{x}'_{k+1} \end{array} \right] = \mathbf{C}_1^{'-1} \left[\begin{array}{c} \mathbf{y}_k \\ \mathbf{0}_{p\times 1} \end{array} \right] + \mathbf{v}'$$

As we can see, this algorithm requires per iteration 2 FFTs of size $n+p$ and 2 FFTs of size $2(n+p)$, which is 25% less than the CGA when p is small (which is usually the case in practice).

Now, an important question is: what are the optimum values for the parameters β and p (for the fastest convergence rate of the method)? We do not have an answer to this question. However, we have empirical evidence from many simulations, that the following procedure yields a good choice.

Consider the function

$$f(l) = \sum_{j=1}^{l} a_j, \; l = 1, 2, ..., p. \tag{14}$$

Many simulations show that the l corresponding to the first minimum of function $f(l)$ is a good candidate for the optimum p. Of course, if all the a_j ($j = 1, 2, ..., p$) are positive, we have this minimum for $l = 1$. In this case, in practice, an $l = 1$ or a bit greater is a good choice.

A good choice for β associated with the parameter $p \neq 0$ seems to be

$$\beta = \beta_0 \sqrt{\sum_{j=1}^{p} a_j^2}, \tag{15}$$

where $\beta_0 \geq 1$ (actually, $\beta_0 \approx 1$).

4 Simulations

The success of our algorithm can only be judged by using it on many different matrices. Still, it is worthwhile to illustrate its performance on one example. There is one matrix that is used very often by other authors [7], [8], [9] to compare iterative methods, so we use it as well. The entries down the lth diagonal of this matrix $\mathbf{A}$ are $a_l = (1+l)^{-s}$, $l = 0, 1, ..., n$. For $s = 2$ the entries decrease quickly and the sum of a_l converges. In this case, all the algorithms perform well. For $s \leq 1$ the sum is divergent. In our example we use $s = 0.3$, which is a difficult case because $\mathbf{A}$ is close to a positive semi-definite matrix. Also, for the right-hand side $\mathbf{b}$ of (1) we take randomly chosen entries from a uniform distribution over $(0, 1)$. A convenient measure of how accurately the vector $\mathbf{x}_k$ satisfies (1) is the Euclidean norm of the residual $\mathbf{r}_k = \mathbf{A}\mathbf{x}_k - \mathbf{b}$. We propose to use the relative residual in dB defined as follows:

$$rr = 20 \log_{10}(\frac{\|\mathbf{r}_k\|_2}{\|\mathbf{b}\|_2}). \tag{16}$$

Figure 1 shows, for $n = 2048$, the behavior of the CGA, the simple version, and the proposed improved version. We can see that the improved version (with $p = 6$) outperforms the two other algorithms. Figures 2 and 3 show how the CGA and the new improved method converge with different sizes ($n = 512, 1024, 2048$) of the linear system. While the convergence of the CGA (Fig. 2) clearly depends on n, it is striking that it is not much the case for the new improved method (Fig. 3). Many other simulations (not shown here) lead us to believe that with the optimum values for β and p, the proposed algorithm converges much faster than the CGA.

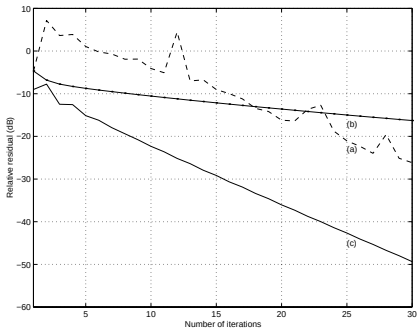

Figure 1: Convergence of the CGA (a), the simple method (b), and the proposed improved method (c), with a linear system of size $n = 2048$.

5 Conclusion

We have proposed a new iterative method which, at least in spirit, is similar to the basic iterative algorithms. We have shown how to apply the basic iterative algorithms to a Toeplitz system and proposed a much faster version by just augmenting the size of the system. Many simulations have shown that the proposed method outperforms the CGA and that the rate of its convergence depends very little on the size

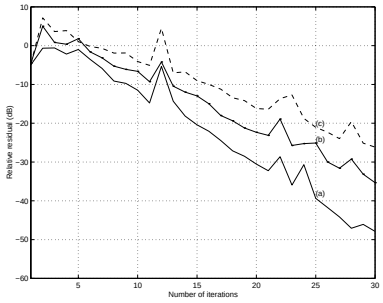

Figure 2: Convergence of the CGA with different sizes of the linear system; $n = 512$ (a), $n = 1024$ (b), and $n = 2048$ (c).

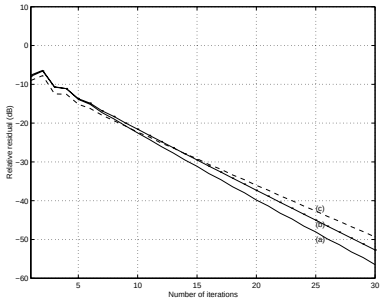

Figure 3: Convergence of the proposed improved method with different sizes of the linear system; $n = 512$ (a), $n = 1024$ (b), and $n = 2048$ (c).

n of the system. However, the dependence of the rate of convergence on the two parameters p and β needs further study.

References

[1] G. H. Golub and C. F. Van Loan, *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 1996.

[2] D. M. Young, *Iterative Solution of Large Linear Systems*. Academic Press Inc., NY, 1971.

[3] R. W. Freund, G. H. Golub, and N. M. Nachtigal, "Iterative solution of linear systems," *Acta Numerica*, Vol. 1, pp. 57-100, 1992.

[4] O. Axelsson, *Iterative Solution Methods*. Cambridge University Press, Melbourne, 1994.

[5] Y. Saad, *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, Boston, 1996.

[6] L. A. Hageman and D. M. Young, *Applied Iterative Methods*. Academic Press Inc., NY, 1981.

[7] G. Strang, "A proposal for Toeplitz matrix calculations," *Studies in Appl. Math.*, 74, pp. 171-176, 1986.

[8] T. F. Chan, "An optimal circulant preconditioner for Toeplitz systems," *SIAM J. Sci. Stat. Comp.*, Vol. 9, No. 4, pp. 766-771, Jul. 1988.

[9] R. H. Chan and G. Strang, "Toeplitz equations by conjugate gradients with circulant preconditioner," *SIAM J. Sci. Stat. Comp.*, Vol. 10, No. 1, pp. 104-119, Jan. 1989.